

```
; INTELLEC8 MONITOR, 8080 VERSION
;
; COPYRIGHT (C) 1973
; INTEL CORPORATION
; 3065 BOWERS AVENUE
; SANTA CLARA, CALIFORNIA 95051
;
; <LEGAL COMMAND> ::= <ASSIGN I/O COMMAND>
;                   <BNPF PUNCH COMMAND>
;                   <COMPARE COMMAND>
;                   <DISPLAY MEMORY COMMAND>
;                   <ENDFILE COMMAND>
;                   <FILL MEMORY COMMAND>
;                   <PROGRAM EXECUTE COMMAND>
;                   <HEXADECIMAL ARITHMETIC COMMAND>
;                   <LOAD BNPF COMMAND>
;                   <MOVE MEMORY COMMAND>
;                   <LEADER COMMAND>
;                   <PROGRAM COMMAND>
;                   <READ HEXADECIMAL FILE COMMAND>
;                   <SUBSTITUTE MEMORY COMMAND>
;                   <TRANSFER COMMAND>
;                   <WRITE HEXADECIMAL RECORD COMMAND>
;                   <REGISTER MODIFY COMMAND>
;
; <ASSIGN I/O COMMAND> ::= A<LOGICAL DEVICE>=<PHYSICAL DEVICE>
;
; <BNPF PUNCH COMMAND> ::= B<NUMBER>,<NUMBER>
;
; <COMPARE COMMAND> ::= C<NUMBER>
;
; <DISPLAY MEMORY COMMAND> ::= D<NUMBER>,<NUMBER>
;
; <ENDFILE COMMAND> ::= E<NUMBER>
;
; <FILL MEMORY COMMAND> ::= F<NUMBER>,<NUMBER>,<NUMBER>
;
; <PROGRAM EXECUTE COMMAND> ::= G<NUMBER>,<NUMBER>,<NUMBER>
;
; <HEXADECIMAL ARITHMETIC COMMAND> ::= H<NUMBER>,<NUMBER>
;
; <LOAD BNPF COMMAND> ::= L<NUMBER>,<NUMBER>
;
; <MOVE MEMORY COMMAND> ::= M<NUMBER>,<NUMBER>,<NUMBER>
;
; <LEADER COMMAND> ::= N
;
; <PROGRAM COMMAND> ::= P<NUMBER>,<NUMBER>,<NUMBER>
;
; <READ HEXADECIMAL FILE COMMAND> ::= R<NUMBER>
;
```



```

; <SUBSTITUTE MEMORY COMMAND> ::= S<NUMBER>
;
; <TRANSFER COMMAND> ::= T<NUMBER>
;
; <WRITE HEXADECEMAL RECORD COMMAND> ::= W<NUMBER>,<NUMBER>
;
; <REGISTER MODIFY COMMAND> ::= X<REGISTER IDENTIFIER>
;
; <LOGICAL DEVICE> ::= CONSOLE|READER|LIST|PUNCH
;
; <PHYSICAL DEVICE> ::= CRT|TTY|PTR|PTP|LPT
;
; <REGISTER IDENTIFIER> ::= A|B|C|D|E|F|I|H|L|M|P|S
;
; <NUMBER> ::=          <HEX DIGIT>
;                  <NUMBER><HEX DIGIT>
;
; <HEX DIGIT> ::= 0|1|2|3|4|5|6|7|8|9|A|B|C|D|E|F
;
; SYSTEM SIGNS ON WITH <CR><LF><,>
;

```

000A

```

VER EQU 10 ; VERSION 1.0
TITLE ' 8080 MONITOR, VERSION 1.0

```

```

; I/O DEVICE OUTPUT COMMAND PORT 1 (TTC) BIT VALUES
;

```

BIT	REST	MNEMONIC	DESCRIPTION
0	0	RBIT	TTY READER GO/NO GO
1	0	PCMD	PTP GO/NO GO
2	0	RCMD	PTR GO/NO GO
3	1	DSB	PROM ENABLE/DISABLE, DSB=1
4	0		DATA IN T/C
5	0		DATA OUT T/C
6	0	PHIT	1702 PROM PROG. GO/NO GO
7	0	PBITA	1702A PROM PROG. GO/NO GO

```

; I/O DEVICE INPUT STATUS PORT 1 (TTS) BIT VALUES
;

```

BIT	REST	MNEMONIC	DESCRIPTION
0	1	TTYDA	IF TTYDA = 0, INPUT IS READY
1	1		OVERRUN ERROR
2	0	TTYBE	IF TTYBE = 0, OUTPUT IS READY
3	1		FRAMING ERROR
4	1		PARITY ERROR
5	0	PTRDA	IF PTRDA = 1, PTR HAS CHAR
6	1	PRDY	IF PRDY = 1, PTP IS READY
7			UNASSIGNED

```

; I/O DEVICE INPUT STATUS PORT 5 (CRTS) BIT VALUES
;

```

```

;
;      BIT      REST      MNEMONIC      DESCRIPTION
;
;      0        1        CRTDA          IF CRTDA = 0, INPUT IS READY
;      1        1              OVERRUN ERROR
;      2        0        CRTBE          IF CRTBE = 0, OUTPUT IS READY
;      3        1              FRAMING ERROR
;      4        1              PARITY ERROR
;      5              UNASSIGNED
;      6              UNASSIGNED
;      7              UNASSIGNED
;
; I/O COMMAND CONSTANTS
;
0001      RBIT EQU      1
0002      PCMD EQU      2
0004      RCMD EQU      4
0008      DSB EQU      8
0080      PBITA EQU     80H
;
; TTY I/O CONSTANTS
;
0000      TTI EQU      0      ; TTY INPUT DATA PORT
0000      TTO EQU      0      ; TTY OUTPUT DATA PORT
0001      TTS EQU      1      ; TTY INPUT STATUS PORT
0001      TTC EQU      1      ; TTY OUTPUT COMMAND PORT
0009      TTYGO EQU     RBIT OR DSB ; START TTY READER
0008      TTYNO EQU     DSB      ; STOP TTY READER
0001      TTYDA EQU      1      ; DATA AVAILABLE
0004      TTYBE EQU      4      ; TRANSMIT BUFFER EMPTY
;
; CRT I/O CONSTANTS
;
0004      CRTI EQU      4      ; CRT INPUT DATA PORT
0005      CRTS EQU      5      ; CRT INPUT STATUS PORT
0004      CRTO EQU      4      ; CRT OUTPUT DATA PORT
0001      CRTDA EQU      1      ; DATA AVAILABLE
0004      CRTBE EQU      4      ; TRANSMIT BUFFER EMPTY
;
; PTR I/O CONSTANTS
;
0003      PTRI EQU      3      ; PTR INPUT DATA PORT (NOT INVERTED)
0001      PTRS EQU      TTS      ; PTR INPUT STATUS PORT
0001      PTRC EQU      TTC      ; PTR OUTPUT COMMAND PORT
000C      PTRGO EQU     RCMD OR DSB ; START PTR
0008      PTRNO EQU     TTYNO      ; STOP PTR
0020      PTRDA EQU     20H      ; PTR DATA AVAILABLE
;
; PTP I/O CONSTANTS
;
0003      PTPO EQU      3      ; PTP OUTPUT DATA PORT

```



```

0001      PTPS EQU      TTS           ; PTP INPUT STATUS PORT
0001      PTPC EQU      TTC           ; PTP OUTPUT COMMAND PORT
0040      PRDY EQU      40H          ; PUNCH READY STATUS
000A      PTPGO EQU     PCMD OR DSB   ; START PUNCH
0008      PTPNO EQU     TTYNO        ; STOP PUNCH
;
; FROM PROGRAMMER I/O CONSTANTS
;
0002      PAD EQU       2             ; FROM ADDRESS OUTPUT PORT
0003      PDO EQU      PIPO          ; FROM DATA OUTPUT PORT
0002      PDI EQU       2             ; FROM DATA INPUT PORT
0001      PROMC EQU     TTC           ; PROGRAMMING PULSE OUTPUT PORT
0080      PROGO EQU     PBITA        ; START PROGRAMMING
0000      PRONO EQU     0             ; STOP PROGRAMMING
0000      ENB EQU       0             ; ENABLE PROGRAMMER
;
00FF      LDLY EQU      0FFH          ; COUNTER FOR 520 MS DELAY
0014      DLY EQU       20           ; COUNTER FOR 2.0 MS DELAY
000D      CR EQU        0DH          ; ASCII VALUE OF CARRIAGE RETURN
000A      LF EQU        0AH          ; ASCII VALUE OF LINE FEED
;
; CONDITIONAL ASSEMBLY SWITCHES
;
0000      FALSE EQU     0
FFFF      TRUE EQU      NOT FALSE
0000      DEBUG EQU     FALSE        ; DEBUG MODE -
;                                     ; DISABLE CERTAIN CODE SECTIONS
;                                     ; SO THAT MODIFICATIONS MAY BE DEBUGGED
;
; MACRO DEFINITIONS
;
1          FETCH MACRO  VALUE         ; FETCH THE ADDRESS OF A VALUE
1          LXI          H,VALUE       ; IN THE STACK
1          DAD          SP
          ENDM
;
FFFF      FIRST SET     TRUE
1          MODIO MACRO  TABLE,MASK
1          LXI          H,TABLE       ; ADDRESS OF PHYSICAL UNIT TABLE
1          MVI          B,MASK        ; B = SELECT BIT MASK
2          IF          FIRST          ; EMIT THIS CODE ONCE,
2                                     ; BRANCH TO IT THEREAFTER
2          IOMOD:
2          FIRST SET     FALSE
2          PUSH         B             ; SAVE BC
2          CALL         NOISE        ; SCAN INPUT AND ECHO UNTIL
2                                     ; PHYSICAL DEVICE CHAR IS ENCOUNTERED
2          MVI          C,4          ; SET TABLE LENGTH
2          CALL         TEST         ; COMPARE PHYSICAL DEVICE AGAINST
2                                     ; TABLE, RETURN HL -> BIT PATTERN
2          JC           LER          ; ERROR, INCORRECT PHYSICAL DEVICE

```

```

2          CALL    SCANOUT      ; SCAN INPUT AND ECHO UNTIL CR,LF
2          POP     B
2          MOV     C,M          ; GET DEVICE SELECT BITS
2          LXI     H,I0BYT      ; GET I/O STATUS
2          MOV     A,M
2          ANA     B            ; CLEAR FIELD
2          ORA     C            ; SET NEW STATUS
2          MOV     M,A          ; RETURN TO MEMORY
2          JMP     START
2
2      TEST:      CMP     M      ; INDEX THROUGH PHYSICAL UNIT TABLE
2              INX     H        ; COMPARE DEVICE CHAR WITH LEGAL VALUES
2              RZ
2              INX     H        ; RETURN WITH HL -> DEVICE SELECT BITS
2              DCR     C
2              JNZ     TEST     ; CONTINUE LOOKUP
2              STC
2              RET
2          ENDF
1          IF      NOT FIRST
2              JMP     IOMOD
2          ENDF
1          ENDM

;
; I/O STATUS BYTE MASKS AND VALUES
;
00FC      CMSG    EQU     11111100B    ; MASK FOR CONSOLE I/O
00F3      RMSG    EQU     11110011B    ; MASK FOR READER INPUT
00CF      PMSG    EQU     11001111B    ; MASK FOR PUNCH OUTPUT
003F      LMSG    EQU     00111111B    ; MASK FOR LIST OUTPUT
;
0000      CTTY    EQU     0            ; CONSOLE I/O = TTY
0001      CCRT    EQU     1            ; CONSOLE I/O = CRT
0002      BATCH   EQU     2            ; BATCH MODE,
; INPUT = READER, OUTPUT = LIST
0003      CUSE    EQU     3            ; USER DEFINED CONSOLE I/O
0000      RTTY    EQU     0            ; READER = TTY
0004      RPTR    EQU     4            ; READER = PTR
0008      RUSE1   EQU     8            ; USER DEFINED READER (1)
000C      RUSE2   EQU     0CH          ; USER DEFINED READER (2)
0000      PTTY    EQU     0            ; PUNCH = TTY
0010      PPTP    EQU     10H          ; PUNCH = PTP
0020      PUSE1   EQU     20H          ; USER DEFINED PUNCH (1)
0030      PUSE2   EQU     30H          ; USER DEFINED PUNCH (2)
0000      LTTY    EQU     0            ; LIST = TTY
0040      LCRT    EQU     40H          ; LIST = CRT
0080      LUSE1   EQU     80H          ; LIST = LPT
00C0      LUSE2   EQU     0C0H         ; USER DEFINED LIST
;
; USER DEFINED DEVICE ENTRY POINTS
;

```



```

3700      CILOC EQU      3700H      ; USER CONSOLE INPUT
3703      COLOC EQU      3703H      ; USER CONSOLE OUTPUT
3706      R1LOC EQU      3706H      ; USER READER 1
3709      R2LOC EQU      3709H      ; USER READER 2
370C      P1LOC EQU      370CH      ; USER PUNCH 1
370F      P2LOC EQU      370FH      ; USER PUNCH 2
3712      L1LOC EQU      3712H      ; USER LIST (1)
3715      L2LOC EQU      3715H      ; USER LIST (2)
3718      CSLOC EQU      3718H      ; USER CONSOLE STATUS

;
1          IF          DEBUG
1          ORG          800H          ; LOCATE IN RAM FOR DEBUG
      ENDIF

;
1          IF          NOT DEBUG
3800 1      ORG          3800H          ; LOCATE IN TOP 8 ROMS IN 16K
      ENDIF

; BRANCH TABLE FOR I/O SYSTEM
;
3800      C32738      JMP      BEGIN      ; RESET ENTRY POINT
3803      C3763C      JMP      CI          ; CONSOLE INPUT
3806      C3943E      JMP      RI          ; READER INPUT
3809      C3323C      JMP      CO          ; CONSOLE OUTPUT
380C      C36C3E      JMP      PO          ; PUNCH OUTPUT
380F      C3CC3D      JMP      LO          ; LIST OUTPUT
3812      C3B73C      JMP      CSTS       ; CONSOLE INPUT STATUS
3815      C39D3D      JMP      IOCHK      ; I/O SYSTEM STATUS
3818      C3A13D      JMP      IOSET      ; SET I/O CONFIGURATION
381B      C3E13D      JMP      MEMCK      ; COMPUTE SIZE OF MEMORY

; POINTERS TO RAM
;
0008      RS1 EQU      8          ; PESTART 1 LOGIC
;
; STATUS BYTE FOR I/O SYSTEM
;
1          IF          NOT DEBUG
0003 1      IOBYT EQU      3          ; USE LOCATION 0003H
      ENDIF
1          IF          DEBUG
1          IOBYT EQU      4          ; USE LOCATION 0004H
      ENDIF
0000      INIT EQU      0          ; INITIALLY,
; CONSOLE = TTY,
; READER = TTY,
; PUNCH = TTY,
; LIST = TTY

;
381E      ODOA5645    VERS: DB      CR,LF,'VER '
3822      5220

```

```

      1      IF      NOT DEBUG
3824 1 312E30      DB      VER/10+'0','.',',VER MOD 10+'0'
      ENDIF

      1      IF      DEBUG
      1      DB      'X,X'
      ENDIF
0009      LVER EQU      $-VERS
      ;
      ; PROGRAM ENTRY POINT
      ;
      ; LOCATE THE STACK IN THE TOP OF AVAILABLE RAM
      ;
3827      BEGIN:
3827      210300      LXI      H,IOBYT      ; POINT HL AT IOBYT
382A      3600      MVI      M,INIT      ; INITIAL VALUE OF I/O
382C      2E00      MVI      L,0
382E      BGO:
382E      46      MOV      B,M      ; FETCH DATA FROM RAM
382F      36AA      MVI      M,0AAH      ; PUT TEST VALUE IN PLACE
3831      7E      MOV      A,M      ; FETCH TEST VALUE
3832      70      MOV      M,B      ; RESTORE ORIGINAL DATA
3833      24      INR      H      ; POINT TO LOC+256
3834      FEAA      CPI      0AAH      ; COMPARE FETCHED TEST DATA WITH KNOWN
3836      CA2E38      JZ      BGO      ; EQUAL, STILL IN RAM
3839      25      DCR      H
      1      IF      DEBUG
      1      MVI      H,2      ; SET STACK AT 200H FOR DEBUG
      ENDIF
383A      0612      MVI      B,ENDX-EXIT      ; MOVE EXIT TEMPLATE TO RAM
383C      11AC3F      LXI      D,ENDX
383F      BG1:
383F      1B      DCX      D
3840      1A      LDAX      D
3841      2B      DCX      H
3842      77      MOV      M,A
3843      05      DCR      B
3844      C23F38      JNZ      BG1
3847      F9      SPHL      ; SET STACK
3848      210001      LXI      H,100H
384B      E5      PUSH      H
384C      2600      MVI      H,0
384E      E5      PUSH      H
384F      E5      PUSH      H
3850      E5      PUSH      H      ; PUSH REGISTERS ON STACK
      1      IF      NOT DEBUG      ; IF IN DEBUG MODE, DON'T SET TRAPS
3851 1 3EC3      MVI      A,(JMP RESTART)
3853 1 320800      STA      RS1
3856 1 21F73E      LXI      H,RESTART      ; SET UP RESTART 1 FOR BREAKPOINT
3859 1 220900      SHLD      RS1+1      ; LOGIC
      ENDIF

```



```

;
; TYPE SIGN-ON
;
385C 211E38      LXI      H,VERS      ; ADDRESS OF MESSAGE
385F 1609        MVI      D,LVER      ; LENGTH OF MESSAGE
3861                                VERO:
3861 4E          MOV      C,M
3862 23          INX      H
3863 CD323C      CALL     CO
3866 15          DCR      D
3867 C26138      JNZ      VERO

;
; MAIN COMMAND LOOP
;
386A                                START:
386A FB          EI              ; ENABLE INTERRUPTS
386B 3E08        MVI      A,TTYNO    ; RESET TTY, PTR, PTP,
386D D301        OUT      TTC        ; AND PROM PROGRAMMER
386F CDAD3C      CALL     CRLF       ; TYPE <CR>,<LF>
3872 0E2E        MVI      C,'.'
3874 CD323C      CALL     CO          ; OUTPUT A PERIOD
3877 CD6D3F      CALL     TI          ; GET A CHARACTER
387A D641        SUI      'A'        ; TEST FOR A-X
387C FA6A38      JM       START      ; LT A, ERROR
387F FE18        CPI      'X'-'A'+1
3881 F2203C      JP       LER        ; GT X, ERROR
3884 87          ALD      A          ; *2
3885 219338      LXI      H,TBL      ; ADDRESS OF TABLE
3888 0600        MVI      B,0        ; CLEAR B
388A 4F          MOV      C,A        ; INDEX TO C
388B 09          DAD      B          ; COMPUTE TABLE ADDRESS, PUT IN HL
388C 7E          MOV      A,M        ; GET LSB OF ADDRESS
388D 23          INX      H          ; POINT TO NEXT ADDRESS
388E 66          MOV      H,M        ; GET MSB OF ADDRESS
388F 6F          MOV      L,A        ; LSB TO L
3890 0E02        MVI      C,2        ; C IS SET UP FOR 2 PARAMETER COMMANDS
3892 E9          PCHL              ; BRANCH TO ROUTINE

;
; COMMAND BRANCH TABLE
;
3893                                TBL:
3893 C338        DW       ASSIGN      ; ASSIGN I/O UNITS
3895 1F39        DW       BNPF       ; B = PUNCH BNPF
3897 6839        DW       COMP       ; C = COMPARE PROM WITH MEMORY
3899 9839        DW       DISP       ; D = DISPLAY RAM MEMORY
389B BC39        DW       EOF        ; E = ENDFILE A HEXADECIMAL FILE
389D DE39        DW       FILL       ; F = FILL MEMORY
389F EF39        DW       GOTO       ; G = GO TO MEMORY ADDRESS
38A1 393A        DW       HEXN       ; H = HEXADECIMAL SUM AND DIFFERENCE
38A3 203C        DW       LER        ; I =
38A5 203C        DW       LER        ; J =

```



```

38A7 203C DW LER ; K =
38A9 563A DW LOAD ; L = LOAD BNPF TAPE
38AB 6A3A DW MOVE ; M = MOVE MEMORY
38AD 7D3A DW NULL ; N = PUNCH NULLS FOR LEADER
38AF 203C DW LER ; O =
38B1 833A DW PROG ; P = PROGRAM A 1702A PROM
38B3 203C DW LER ; Q =
38B5 DF3A DW READ ; R = READ HEXADECIMAL FILE
38B7 263B DW SUBS ; S = SUBSTITUTE MEMORY
38B9 543B DW TRAN ; T = TRANSFER A PROM TO MEMORY
38BB 203C DW LER ; U =
38BD 203C DW LER ; V =
38BF 6F3B DW WRITE ; W = WRITE HEX TAPE
38C1 BC3B DW X ; X = EXAMINE AND MODIFY REGISTERS

```

```

;
; PROCESS I/O DEVICE ASSIGNMENT COMMANDS

```

```

38C3 ; ASSIGN:
38C3 CD6D3F CALL TI ; GET LOGICAL DEVICE CHARACTER
38C6 FE43 CPI 'C' ; CONSOLE?
38C8 C2F838 JNZ ASO ; TEST FOR READER
; MODIO
38CB 1 217A3F + LXI H,ICT ; MODIFY CONSOLE DEVICE
38CE 1 06FC + MVI B,CMASK ; ADDRESS OF PHYSICAL UNIT TABLE
2 + IF FIRST ; B = SELECT BIT MASK
2 + ; EMIT THIS CODE ONCE,
; BRANCH TO IT THEREAFTER
38D0 2 +IOMOD:;
0000 2 +FIRST SET FALSE
38D0 2 C5 + PUSH B ; SAVE BC
38D1 2 CD123E + CALL NOISE ; SCAN INPUT AND ECHO UNTIL
2 + ; PHYSICAL DEVICE CHAR IS ENCOUNTERED
38D4 2 0E04 + MVI C,4 ; SET TABLE LENGTH
38D6 2 CDEB38 + CALL TEST ; COMPARE PHYSICAL DEVICE AGAINST
2 + ; TABLE, RETURN HL -> BIT PATTERN
38D9 2 DA203C + JC LER ; ERROR, INCORRECT PHYSICAL DEVICE
38DC 2 CD623F + CALL SCANOUT ; SCAN INPUT AND ECHO UNTIL CR,LF
38DF 2 C1 + POP B
38E0 2 4E + MOV C,M ; GET DEVICE SELECT BITS
38E1 2 210300 + LXI H,IOBYT ; GET I/O STATUS
38E4 2 7E + MOV A,M
38E5 2 A0 + ANA B ; CLEAR FIELD
38E6 2 B1 + ORA C ; SET NEW STATUS
38E7 2 77 + MOV M,A ; RETURN TO MEMORY
38E8 2 C36A38 + JMP START
38EB 2 +TEST:;
38EB 2 BE + CMP M ; INDEX THROUGH PHYSICAL UNIT TABLE
38EC 2 23 + INX H ; COMPARE DEVICE CHAR WITH LEGAL VALUES
38ED 2 C8 + RZ ; RETURN WITH HL -> DEVICE SELECT BITS
38EE 2 23 + INX H
38EF 2 0D + DCR C
38F0 2 C2EB38 + JNZ TEST ; CONTINUE LOOKUP

```



```

38F3 2 37      +      STC                      ; ERROR RETURN
38F4 2 C9      +      RET
1          +      ENDIF
2          +      IF      NOT FIRST
38F5 2 C3D038  +      JMP      IOMOD
1          +      ENDIF

38F8          AS0:
38F8 FE52      CPI      'R'                      ; READER?
38FA C20539    JNZ      AS1                      ; TEST FOR PUNCH
1          +      MODIO  IRT,RMSK                ; MODIFY READER DEVICE
38FD 1 21823F  +      LXI      H,IRT              ; ADDRESS OF PHYSICAL UNIT TABLE
3900 1 06F3    +      MVI      B,RMSK            ; B = SELECT BIT MASK
2          +      IF      FIRST                  ; EMIT THIS CODE ONCE,
2          +                      ; BRANCH TO IT THEREAFTER
2          + IOMOD:
2          + FIRST SET      FALSE
2          +      PUSH     B                      ; SAVE BC
2          +      CALL     NOISE                  ; SCAN INPUT AND ECHO UNTIL
2          +                      ; PHYSICAL DEVICE CHAR IS ENCOUNTERED
2          +      MVI      C,4                    ; SET TABLE LENGTH
2          +      CALL     TEST                  ; COMPARE PHYSICAL DEVICE AGAINST
2          +                      ; TABLE, RETURN HL -> BIT PATTERN
2          +      JC      LER                      ; ERROR, INCORRECT PHYSICAL DEVICE
2          +      CALL     SCANOUT                ; SCAN INPUT AND ECHO UNTIL CR,LF
2          +      POP      B
2          +      MOV      C,M                      ; GET DEVICE SELECT BITS
2          +      LXI      H,I0BYT                ; GET I/O STATUS
2          +      MOV      A,M
2          +      ANA      B                      ; CLEAR FIELD
2          +      ORA      C                      ; SET NEW STATUS
2          +      MOV      M,A                    ; RETURN TO MEMORY
2          +      JMP      START
2          + TEST:
2          +      CMP      M                      ; INDEX THROUGH PHYSICAL UNIT TABLE
2          +      INX      H                      ; COMPARE DEVICE CHAR WITH LEGAL VALUES
2          +      RZ                      ; RETURN WITH HL -> DEVICE SELECT BITS
2          +      INX      H
2          +      DCR      C
2          +      JNZ      TEST                  ; CONTINUE LOOKUP
2          +      STC                      ; ERROR RETURN
2          +      RET
1          +      ENDIF
2          +      IF      NOT FIRST
3902 2 C3D038  +      JMP      IOMOD
1          +      ENDIF

3905          AS1:
3905 FE50      CPI      'P'                      ; PUNCH?
3907 C21239    JNZ      AS2                      ; TEST FOR LIST
1          +      MODIO  OPT,RMSK                ; MODIFY PUNCH DEVICE
390A 1 218A3F  +      LXI      H,OPT              ; ADDRESS OF PHYSICAL UNIT TABLE
390D 1 06CF    +      MVI      B,RMSK            ; B = SELECT BIT MASK

```



```

2      +      IF      FIRST      ; EMIT THIS CODE ONCE,
2      +                                          ; BRANCH TO IT THEREAFTER
2      +IOMOD:;
2      +FIRST SET      FALSE
2      +      PUSH     B          ; SAVE BC
2      +      CALL     NOISE      ; SCAN INPUT AND ECHO UNTIL
2      +                                          ; PHYSICAL DEVICE CHAR IS ENCOUNTERED
2      +      MVI      C,4        ; SET TABLE LENGTH
2      +      CALL     TEST       ; COMPARE PHYSICAL DEVICE AGAINST
2      +                                          ; TABLE, RETURN HL -> BIT PATTERN
2      +      JC       LER        ; ERROR, INCORRECT PHYSICAL DEVICE
2      +      CALL     SCANOUT    ; SCAN INPUT AND ECHO UNTIL CR,LF
2      +      POP      B
2      +      MOV      C,M        ; GET DEVICE SELECT BITS
2      +      LXI      H,IUBYT    ; GET I/O STATUS
2      +      MOV      A,M
2      +      ANA      B          ; CLEAR FIELD
2      +      ORA      C          ; SET NEW STATUS
2      +      MOV      M,A        ; RETURN TO MEMORY
2      +      JMP      START
2      +TEST:
2      +      CMP      M          ; INDEX THROUGH PHYSICAL UNIT TABLE
2      +      INX      H          ; COMPARE DEVICE CHAR WITH LEGAL VALUES
2      +      RZ
2      +      INX      H          ; RETURN WITH HL -> DEVICE SELECT BITS
2      +      DCR      C
2      +      JNZ     TEST       ; CONTINUE LOOKUP
2      +      STC
2      +      RET
2      +      ENDIF
1      +      IF      NOT FIRST
390F 2 C3D038 +      JMP      IOMOD
1      +      ENDIF
3912      AS2:
3912      FE4C      CPI      'L'    ; LIST?
3914      C2203C    JNZ      LER    ; ERROR
1      +      MODIO    OLT,LMSK    ; MODIFY LIST DEVICE
3917 1 21923F    +      LXI      H,OLT ; ADDRESS OF PHYSICAL UNIT TABLE
391A 1 063F      +      MVI      B,LMSK ; B = SELECT BIT MASK
2      +      IF      FIRST      ; EMIT THIS CODE ONCE,
2      +                                          ; BRANCH TO IT THEREAFTER
2      +IOMOD:;
2      +FIRST SET      FALSE
2      +      PUSH     B          ; SAVE BC
2      +      CALL     NOISE      ; SCAN INPUT AND ECHO UNTIL
2      +                                          ; PHYSICAL DEVICE CHAR IS ENCOUNTERED
2      +      MVI      C,4        ; SET TABLE LENGTH
2      +      CALL     TEST       ; COMPARE PHYSICAL DEVICE AGAINST
2      +                                          ; TABLE, RETURN HL -> BIT PATTERN
2      +      JC       LER        ; ERROR, INCORRECT PHYSICAL DEVICE
2      +      CALL     SCANOUT    ; SCAN INPUT AND ECHO UNTIL CR,LF

```

```

2      +      POP      B
2      +      MOV      C,M          ; GET DEVICE SELECT BITS
2      +      LXI      H,IOBYT      ; GET I/O STATUS
2      +      MOV      A,M
2      +      ANA      B          ; CLEAR FIELD
2      +      ORA      C          ; SET NEW STATUS
2      +      MOV      M,A          ; RETURN TO MEMORY
2      +      JMP      START
2      +TEST:
2      +      CMP      M          ; INDEX THROUGH PHYSICAL UNIT TABLE
2      +      INX      H          ; COMPARE DEVICE CHAR WITH LEGAL VALUES
2      +      RZ
2      +      INX      H          ; RETURN WITH HL -> DEVICE SELECT BITS
2      +      DCR      C
2      +      JNZ      TEST      ; CONTINUE LOOKUP
2      +      STC
2      +      RET
1      +      ENDIF
2      +      IF      NOT FIRST
391C 2 C3D038 +      JMP      IOMOD
1      +      ENDIF
;
; PUNCH ROUTINE, PUNCH A BPNF TAPE
;
391F      BPNF:
391F      CD593D      CALL      EXPR          ; GET TWO ADDRESSES
3922      CDAD3C      CALL      CRLF
3925      CDC03D      CALL      LEAD
3928      D1          POP      D          ; GET HIGH ADDRESS
3929      E1          POP      H          ; GET LOW ADDRESS
392A      BNO:
392A      E5          PUSH      H
392B      D5          PUSH      D
392C      CD623E      CALL      PEOL          ; PUNCH CR,LF
392F      0620      MVI      B," "          ; ZERO SUPPRESSION CHARACTER
3931      111027      LXI      D,10000      ; PUNCH ADDRESS IN DECIMAL
3934      CD173D      CALL      DIGIT
3937      11E803      LXI      D,1000
393A      CD173D      CALL      DIGIT
393D      116400      LXI      D,100
3940      CD173D      CALL      DIGIT
3943      1E0A      MVI      E,10
3945      CD173D      CALL      DIGIT
3948      1E01      MVI      E,1
394A      0630      MVI      B,"0"          ; FORCE AT LEAST 1 ZERO
394C      CD173D      CALL      DIGIT
394F      0E20      MVI      C," "
3951      CD6C3E      CALL      PD
3954      D1          POP      D
3955      E1          POP      H
3956      BNI:

```



```

3956 CD363D      CALL    ENCODE      ; ENCODE A MEMORY BYTE INTO BNPF
3959 CD8D3D      CALL    HILO
395C DA7D3A      JC      NULL
395F 7D          MOV     A,L
3960 E603        ANI     03H          ; PUNCH CR,LF,ADDRESS ON MULTIPLE OF 4
3962 C25639      JNZ     BN1
3965 C32A39      JMP     BN0

```

```

;
; COMPARE PROM WITH RAM
;

```

```

3968      COMP:
3968 0D          DCR     C
3969 CD593D      CALL    EXPR          ; GET ONE ADDRESS
396C E1          POP     H              ; LOAD HL
396D 1E00        MVI     E,0          ; COUNT/PROM ADDRESS
396F      CM0:
396F 3E00        MVI     A,ENB
3971 D301        OUT     PROMC
3973 7B          MOV     A,E
3974 2F          CMA
3975 D302        OUT     PAD
3977 DB02        IN      PDI          ; GET PROM DATA
3979 2F          CMA
397A BE          CMP     M
397B CA9339      JZ      CM1
397E F5          PUSH    PSW
397F CDAD3C      CALL    CRLF
3982 CDA83D      CALL    LADR          ; PRINT MEMORY ADDRESS
3985 CD303C      CALL    BLK
3988 7E          MOV     A,M
3989 CDB03D      CALL    LBYTE          ; PRINT RAM DATA
398C CD303C      CALL    BLK
398F F1          POP     PSW
3990 CDB03D      CALL    LBYTE          ; PRINT PROM DATA
3993      CM1:
3993 23          INX     H
3994 1C          INR     E
3995 C26F39      JNZ     CM0
3998 C36A38      JMP     START

```

```

;
; DISPLAY MEMORY IN HEX ON TELEPRINTER
;

```

```

399B      DISP:
399B CD593D      CALL    EXPR          ; GET TWO ADDRESSES
399E D1          POP     D              ; GET HIGH ADDRESS
399F E1          POP     H              ; GET LOW ADDRESS
39A0      DI0:
39A0 CDAD3C      CALL    CRLF
39A3 CDA83D      CALL    LADR          ; PRINT MEMORY ADDRESS
39A6      DI1:
39A6 CD303C      CALL    BLK          ; PRINT SPACE

```

```

39A9 7E      MOV      A,M
39AA CDB03E   CALL     LBYTE      ; PRINT DATA
39AD CD8D3D   CALL     HILO      ; TEST FOR COMPLETION
39B0 DA6A38   JC       START
39B3 7D      MOV      A,L
39B4 E60F     ANI      0FH      ; PRINT CR,LF,ADDRESS ON MULTIPLE OF 16
39B6 C2A639   JNZ      DI1
39B9 C3A039   JMP      DIO

```

```

;
; END OF FILE COMMAND
;

```

```

39BC      EOF:
39BC 0D      DCR      C          ; GET ONE PARAMETER
39BD CD593D   CALL     EXPR
39C0 CD623E   CALL     PEOL      ; PUNCH CR,LF
39C3 0E3A     MVI      C,'I'
39C5 CD6C3E   CALL     PC
39C8 AF      XRA      A          ; CLEAR CHECKSUM
39C9 57      MOV      D,A
39CA CD2B3E   CALL     PBYTE     ; OUTPUT RECORD LENGTH
39CD E1      POP      H
39CE CD233E   CALL     PADR      ; PUNCH EXECUTION ADDRESS
39D1 3E01     MVI      A,1      ; RECORD TYPE 1
39D3 CD2B3E   CALL     PBYTE
39D6 AF      XRA      A
39D7 92      SUB      D          ; OUTPUT CHECKSUM
39D8 CD2B3E   CALL     PBYTE
39DB C37D3A   JMP      NULL      ; PUNCH TRAILER AND RETURN

```

```

;
; FILL ROUTINE, FILL RAM MEMORY BLOCK WITH CONSTANT
;

```

```

39DE      FILL:
39DE 0C      INR      C          ; GET 3 PARAMETERS
39DF CD593D   CALL     EXPR
39E2 C1      POP      B          ; GET DATA IN C
39E3 D1      POP      D          ; GET HIGH ADDRESS
39E4 E1      POP      H          ; GET LOW ADDRESS
39E5      FIO:
39E5 71      MOV      M,C          ; STORE CONSTANT IN MEMORY
39E6 CD8D3D   CALL     HILO      ; TEST FOR COMPLETION
39E9 D2E539   JNC      FIO      ; CONTINUE LOOPING
39EC C36A38   JMP      START

```

```

;
; GO TO <ADDRESS>, OPTIONALLY SET TRAPS
;

```

```

39EF      GOTO:
39EF CD463E   CALL     PCHK      ; GET A CHARACTER
39F2 DA313A   JC       GO3      ; CR ENTERED, EXIT
39F5 CA093A   JZ       GO0      ; DON'T MODIFY PC
39F8 CD853D   CALL     EXF      ; GET NEW PC VALUE
39FB D1      POP      D

```



```

1      +      FETCH      PLOC
39FC 1 211300 +      LXI      H,PLOC      ; IN THE STACK
39FF 1 39      +      DAD      SP
3A00      72      MOV      M,D      ; STORE MODIFIED PC IN RAM
3A01      2B      DCX      H
3A02      73      MOV      M,E
3A03      78      MOV      A,B      ; RETRIEVE DELIMITER CHARACTER
3A04      FE0D     CPI      CR
3A06      CA313A   JZ      GO3      ; NO TRAPS TO BE SET
3A09
3A09      1602     GU0:      MVI      D,2      ; SET MAXIMUM OF TWO TRAPS
1      +      FETCH      TLOC
3A0B 1 211400 +      LXI      H,TLOC      ; IN THE STACK
3A0E 1 39      +      DAD      SP
3A0F
3A0F      E5      GU1:      PUSH     H      ; SAVE ADDRESS OF TRAP AREA
3A10      0E01     MVI      C,1
3A12      CD593D   CALL     EXPR      ; GET A TRAP ADDRESS
3A15      58      MOV      E,B      ; SAVE DELIMITER CHARACTER
3A16      C1      POP      B      ; GET ADDRESS IN BC
3A17      E1      POP      H
3A18      78      MOV      A,B
3A19      B1      ORA      C
3A1A      CA273A   JZ      GO2      ; DON'T ALLOW A TRAP AT 0
3A1D      71      MOV      M,C      ; PUT TRAP ADDRESS AWAY
3A1E      23      INX      H
3A1F      70      MOV      M,B
3A20      23      INX      H
3A21      0A      LDAX     B      ; FETCH OP CODE
3A22      77      MOV      M,A      ; PUT IN TRAP AREA
3A23      23      INX      H
3A24      3ECF     MVI      A,(RST 1) ; RESTART 1
3A26      02      STAX     B      ; SET TRAP IN MEMORY
3A27
3A27      7B      GU2:      MOV      A,E      ; TEST DELIMITER CHARACTER
3A28      FE0D     CPI      CR
3A2A      CA313A   JZ      GO3      ; ALL DONE
3A2D      15      DCR      D
3A2E      C20F3A   JNZ     GO1      ; GO GET NEXT TRAP
3A31
3A31      CDAD3C   GU3:      CALL     CRLF
1      +      FETCH      8
3A34 1 210800 +      LXI      H,00008H      ; IN THE STACK
3A37 1 39      +      DAD      SP
3A38      E9      PCHL      ; TAKE THE BRANCH
;
; COMPUTE HEXADECIMAL SUM AND DIFFERENCE
;
3A39
3A39      CD593D   HEXN:      CALL     EXPR      ; GET TWO NUMBERS
3A3C      D1      POP      D

```

```

3A3D  E1          POP      H
3A3E  CDAD3C      CALL     CRLF
3A41  E5          PUSH     H
3A42  19          DAD      D          ; COMPUTE HL+DE
3A43  CDA83D      CALL     LADR       ; DISPLAY SUM
3A46  CD303C      CALL     BLK        ; TYPE A SPACE
3A49  E1          POP      H
3A4A  7D          MOV      A,L        ; COMPUTE HL-DE
3A4B  93          SUB      E
3A4C  6F          MOV      L,A
3A4D  7C          MOV      A,H
3A4E  9A          SBB      D
3A4F  67          MOV      H,A
3A50  CDA83D      CALL     LADR       ; DISPLAY DIFFERENCE
3A53  C36A38      JMP      START

;
; LOAD ROUTINE, LOAD A BNPF TAPE INTO RAM MEMORY
;
LOAD:
3A56                      CALL     EXPR          ; GET TWO ADDRESSES
3A56  CD593D      CALL     CRLF
3A59  CDAD3C      CALL     CRLF
3A5C  D1          POP      D          ; GET HIGH ADDRESS
3A5D  E1          POP      H          ; GET LOW ADDRESS
3A5E                      LOO:
3A5E  CDDC3C      CALL     DECODE       ; CONVERT BNPF, STORE IN MEMORY
3A61  CD8D3D      CALL     HILO        ; TEST FOR COMPLETION
3A64  D25E3A      JNC      LOO         ; KEEP GOING
3A67  C36A38      JMP      START

;
; MOVE A BLOCK OF RAM MEMORY
;
MOVE:
3A6A                      INR      C          ; GET THREE ADDRESSES
3A6A  0C          CALL     EXPR
3A6B  CD593D      CALL     EXPR
3A6E  C1          POP      B          ; DESTINATION
3A6F  D1          POP      D          ; SOURCE END
3A70  E1          POP      H          ; SOURCE BEGIN
3A71                      MVO:
3A71  7E          MOV      A,M          ; GET A DATA BYTE
3A72  02          STAX     B          ; STORE AT DESTINATION
3A73  03          INX      B          ; MOVE DESTINATION POINTER
3A74  CD8D3D      CALL     HILO        ; TEST FOR COMPLETION
3A77  D2713A      JNC      MVO
3A7A  C36A38      JMP      START

;
; PUNCH LEADER/TRAILER
;
NULL:
3A7D                      CALL     LEAD
3A7D  CDC03D      CALL     LEAD
3A80  C36A38      JMP      START
;

```



```

; PROGRAM A 1702A PROM
!
PROG:
3A83      0C          INR      C
3A84      CD593D     CALL     EXPR      ; HL = TOP AFTER RETURN
3A87      CDAD3C     CALL     CRLF
3A8A      C1         POP      B          ; C <- PROM ADDRESS
3A8B      D1         POP      D          ; HIGH ADDRESS
3A8C      E1         POP      H          ; LOW ADDRESS
3A8D
3A8D      0603       PR0:     MVI      B,3      ; RETRY COUNT
3A8F
3A8F      3E00       PR1:     MVI      A,ENB
3A91      D301       OUT      PROMC      ; ENABLE PROM PROGRAMMER
3A93      79        MOV      A,C
3A94      D3FF       OUT      OFFH      ; DISPLAY ADDRESS
3A96      2F        CMA
3A97      D302       OUT      PAD        ; FROM ADDRESS
3A99      DB02       IN       PDI
3A9B      2F        CMA
3A9C      BE        CMP      M
3A9D      CAD53A     JZ       PR2      ; DON'T HAVE TO PROGRAM THE LOC
3AA0      7E        MOV      A,M
3AA1      2F        CMA
3AA2      D303       OUT      PDO        ; OUTPUT DATA
3AA4      3E80       MVI      A,PROG0
3AA6      D301       OUT      PROMC      ; PULSE IT
3AA8      CD563E     CALL     PDLY      ; DELAY 520 MSEC
3AAB      3E00       MVI      A,PROND
3AAD      D301       OUT      PROMC      ; CLEAR PULSE
3AAF      CD053D     CALL     DELAY      ; DELAY 2.0 MSEC.
3AB2      CD053D     CALL     DELAY      ; DELAY 2.0 MSEC.
3AB5      CD053D     CALL     DELAY      ; DELAY 2.0 MSEC.
3AB8      DB02       IN       PDI
3ABA      2F        CMA
3ABB      BE        CMP      M
3ABC      CAD53A     JZ       PR2      ; COMPARE OK
3ABF      C5        PUSH     B
3AC0      0E24       MVI      C,'S'
3AC2      CD323C     CALL     CO
3AC5      C1         POP      B
3AC6      05        DCR      B
3AC7      C28F3A     JNZ      PR1
3ACA      41        MOV      B,C
3ACB      CD303C     CALL     BLK      ; OUTPUT A SPACE
3ACE      78        MOV      A,B      ; DISPLAY PROM ADDRESS
3ACF      CDB03D     CALL     LBYTE
3AD2      C3203C     JMP      LER      ; BAD PROM, ABORT
3AD5
3AD5      0C          PR2:     INR      C          ; INCREMENT PROM ADDRESS
3AD6      CD8D3D     CALL     HILO

```

```

3AD9 D28D3A JNC PRO
3ADC C36A38 JMP START

;
; READ ROUTINE, READ A BINARY TAPE
;
3ADF READ:
3ADF 0D DCR C ; GET ONE ADDRESS
3AE0 CD593D CALL EXPR
3AE3 RED0:
3AE3 E1 POP H ; GET BIAS ADDRESS
3AE4 E5 PUSH H
3AE5 CDEE3E CALL RIX
3AE8 063A MVI B,';'
3AEA 90 SUB B
3AEB C2E33A JNZ RED0 ; SCAN TO RECORD MARK
3AEE 57 MOV D,A ; CLEAR CHECKSUM
3AEF CD5F3C CALL BYTE
3AF2 CA153B JZ RED2 ; ZERO RECORD LENGTH, ALL DONE
3AF5 5F MOV E,A ; E ← RECORD LENGTH
3AF6 CD5F3C CALL BYTE ; GET MSB OF LOAD ADDRESS
3AF9 F5 PUSH PSW ; SAVE IT
3AFA CD5F3C CALL BYTE ; GET LSB OF LOAD ADDRESS
3AFD C1 POP B ; RETRIEVE MSB, PUT IN B
3AFE 4F MOV C,A
3AFF 09 DAD B ; BIAS ADDRESS + LOAD ADDRESS => HL
3B00 CD5F3C CALL BYTE ; RECORD TYPE
3B03 RED1:
3B03 CD5F3C CALL BYTE ; READ DATA
3B06 77 MOV M,A ; PUT IN MEMORY
3B07 23 INX H
3B08 1D DCR E
3B09 C2033B JNZ RED1 ; LOOP UNTIL DONE
3B0C CD5F3C CALL BYTE ; READ CHECKSUM
3B0F C2203C JNZ LER ; CHECKSUM ERROR
3B12 C3E33A JMP RED0 ; GET ANOTHER RECORD
3B15 RED2:
3B15 CD5F3C CALL BYTE ; GET MSB OF TRANSFER ADDRESS
3B18 67 MOV H,A
3B19 CD5F3C CALL BYTE
3B1C 6F MOV L,A
3B1D B4 ORA H
3B1E CA223B JZ RED3 ; IF TRANSFER ADDRESS = 0, RETURN TO KB
3B21 E9 PCHL
3B22 RED3:
3B22 E1 POP H
3B23 C36A38 JMP START

;
; SUBSTITUTE ROUTINE, MODIFY RAM MEMORY WITH KEYBOARD INPUTS
;
3B26 SUBS:
3B26 0D DCR C

```



```

3B27 CD593D      CALL    EXPR          ; GET ONE ADDRESS
3B2A CD493E      CALL    P2C
3B2D DA6A38      JC      START
3B30 E1          POP      H
3B31              SU0:
3B31 7E          MOV      A,M
3B32 CDB03D      CALL    LBYTE          ; DISPLAY DATA
3B35 0E2D        MVI      C,'-'
3B37 CD323C      CALL    CO
3B3A CD463E      CALL    PCHK
3B3D DA6A38      JC      START          ; CR ENTERED, RETURN TO COMMAND MODE
3B40 CA503B      JZ      SU1            ; SPACE ENTERED, SPACE BY
3B43 E5          PUSH     H              ; SAVE MEMORY ADDRESS
3B44 CD853D      CALL    EXF            ; GET NEW VALUE
3B47 D1          POP      D              ; E = VALUE
3B48 E1          POP      H              ; RESTORE MEMORY ADDRESS
3B49 73          MOV      M,E            ; STORE NEW VALUE
3B4A 78          MOV      A,B            ; TEST DELIMITER
3B4B FE0D        CPI      CR
3B4D CA6A38      JZ      START          ; CR ENTERED AFTER LAST SUBSTITUTION
3B50              SU1:
3B50 23          INX      H
3B51 C3313B      JMP      SU0

;
; TRANSFER CONTENTS OF A PROM TO MEMORY
;
3B54              TRAN:
3B54 0D          DCR      C
3B55 CD593D      CALL    EXPR          ; GET ONE ADDRESS
3B58 E1          POP      H              ; HL = MEM ADR
3B59 1E00        MVI      E,0          ; COUNT/PROM ADDRESS
3B5B              TRO:
3B5B 3E00        MVI      A,ENB
3B5D D301        OUT      PROMC          ; ENABLE PROM PROGRAMMER
3B5F 7B          MOV      A,E
3B60 2F          CMA
3B61 D302        OUT      PAD            ; INVERT ADDRESS
3B63 DB02        IN       PDI            ; SET PROM ADDRESS
3B65 2F          CMA                    ; GET PROM DATA
3B66 77          MOV      M,A            ; PUT IN MEMORY
3B67 23          INX      H              ; BUMP MEMORY POINTER
3B68 1C          INR      E              ; BUMP PROM POINTER
3B69 C25B3B      JNZ      TRO            ; GET ANOTHER BYTE
3B6C C36A38      JMP      START

;
; WRITE ROUTINE, WRITE A BINARY TAPE
;
3B6F              WRITE:
3B6F CD593D      CALL    EXPR          ; GET TWO ADDRESSES
3B72 CDAD3C      CALL    CRLF
3B75 D1          POP      D              ; GET HIGH ADDRESS

```

```

3B76 E1 POP H ; GET LOW ADDRESS
3B77 WR10: MOV A,L
3B78 C610 ADI 16
3B7A 4F MOV C,A
3B7B 7C MOV A,H
3B7C CE00 ACI 0
3B7E 47 MOV B,A
3B7F 7B MOV A,E
3B80 91 SUB C
3B81 4F MOV C,A
3B82 7A MOV A,D
3B83 9B SBB B
3B84 DA8C3B JC WRI1 ; RECORD LENGTH = 16
3B87 3E10 MVI A,16
3B89 C38F3B JMP WRI2
3B8C WRI1: MOV A,C ; LAST RECORD
3B8D C611 ADI 17
3B8F WRI2: ORA A
3B90 CA6A38 JZ START
3B93 D5 PUSH D ; SAVE HIGH ADDRESS
3B94 5F MOV E,A ; E = LENGTH
3B95 1600 MVI D,0 ; CLEAR CHECKSUM
3B97 CD623E CALL PEOB ; PUNCH CR,LF
3B9A 0E3A MVI C,' '
3B9C CD6C3E CALL PO
3B9F 7B MOV A,E
3BA0 CD2B3E CALL PBYTE ; PUNCH LENGTH
3BA3 CD233E CALL PADR ; PUNCH ADDRESS
3BA6 AF XRA A
3BA7 CD2B3E CALL PBYTE ; PUNCH RECORD TYPE
3BAA WRI3: MOV A,M
3BAB 23 INX H
3BAC CD2B3E CALL PBYTE ; PUNCH DATA
3BAF 1D DCR E ; DECREMENT LENGTH
3BB0 C2AA3B JNZ WRI3 ; CONTINUE LOOPING
3BB3 AF XRA A
3BB4 92 SUB D
3BB5 CD2B3E CALL PBYTE ; PUNCH CHECKSUM
3BB8 D1 POP D ; RESTORE HIGH ADDRESS
3BB9 C3773B JMP WRI0
;
; EXAMINE AND MODIFY CPU REGISTERS
;
3BBC X1: CALL TI ; GET REGISTER IDENTIFIER
3BBF 21AC3F LXI H,ACTBL ; POINT TO ACCESS TABLE
3BC2 X0:

```


3BC2	8E		CMP	M	
3BC3	CAD33B		JZ	X1	
3BC6	F5		PUSH	PSW	; SAVE CHARACTER
3BC7	7E		MOV	A,M	; NOT THE RIGHT REGISTER
3BC8	B7		ORA	A	
3BC9	FA203C		JM	LER	; END OF TABLE
3BCC	23		INX	H	
3BCD	23		INX	H	
3BCE	23		INX	H	
3BCF	F1		POP	PSW	; RETRIEVE CHARACTER
3BD0	C3C23B		JMP	X0	
3BD3		X1:			
3BD3	CD303C		CALL	BLK	
3BD6		X2:			
3BD6	23		INX	H	
3BD7	7E		MOV	A,M	; DISPLACEMENT
3BD8	EB		XCHG		; SAVE HL IN DE (HL = POINTER TO ACTBL)
3BD9	6F		MOV	L,A	
3BDA	2600		MVI	H,0	
3BDC	39		DAD	SP	
3BDD	EB		XCHG		; RESTORE HL
3BDE	23		INX	H	
3BDF	46		MOV	B,M	; PRECISION;
3BE0	23		INX	H	; POINT AT NEXT REGISTER IDENTIFIER
3BE1	1A		LDAX	D	; 8/16 BIT DISPLAY AND MODIFICATION
3BE2	CDB03D		CALL	LBYTE	; MSB OF 16 BIT REG, ALL OF 8 BIT REG
3BE5	05		DCR	B	
3BE6	CAEE3B		JZ	X3	; 8 BIT DISPLAY
3BE9	1B		DCX	D	
3BEA	1A		LDAX	D	
3BEB	CDB03D		CALL	LBYTE	; LSB OF 16 BIT REG
3BEE		X3:			
3BEE	04		INR	B	
3BEF	0E2D		MVI	C,'-'	
3BF1	CD323C		CALL	CO	
3BF4	CD463E		CALL	PCHK	; SKIP IF NULL ENTRY
3BF7	DA6A38		JC	START	; CR ENTERED, RETURN TO COMMAND MODE
3BFA	CA123C		JZ	X5	
3BFD	E5		PUSH	H	; SAVE POINTER TO ACTBL
3BFE	C5		PUSH	B	; SAVE PRECISION
3BFF	CD853D		CALL	EXF	; GET NEW REG VALUE
3C02	E1		POP	H	
3C03	F1		POP	PSW	; A = PRECISION
3C04	C5		PUSH	B	; B = DELIMITER CHAR
3C05	F5		PUSH	PSW	; A = PRECISION
3C06	7D		MOV	A,L	
3C07	12		STAX	D	; STORE LSB IN REGISTER AREA
3C08	C1		POP	B	; RETRIEVE PRECISION
3C09	05		DCR	B	
3C0A	CA103C		JZ	X4	; 8 BITS ONLY
3C0D	13		INX	D	

```

3C0E 7C      MOV    A,H
3C0F 12      STAX   D          ; STORE MSB IN REGISTER AREA
3C10      X4:
3C10 C1      POP    B          ; RETRIEVE DELIMITER CHAR
3C11 E1      POP    H
3C12      X5:
3C12 7E      MOV    A,M        ; TEST FOR END OF TABLE
3C13 B7      ORA    A
3C14 FA6A38  JM     START
3C17 78      MOV    A,B        ; TEST DELIMITER
3C18 FE0D      CPI    CR
3C1A CA6A38  JZ     START
3C1D C3D63B  JMP    X2
;
; ERROR EXIT
;
3C20      LER:
3C20 CDEA3D  CALL   MEMSIZ      ; COMPUTE TOP OF MEMORY
3C23 11F8FF  LXI    D,-8
3C26 19      DAD    D
3C27 F9      SPHL
3C28 0E2A      MVI    C,'*'    ; RESET STACK POINTER ON ERROR RETURN
3C2A CD323C  CALL   CO
3C2D C36A38  JMP    START
;
; SUBROUTINES
;
3C30      BLK:
3C30 0E20      MVI    C,' '    ; PRINT A BLANK
;
; EXTERNALLY REFERENCED ROUTINE
; CONSOLE OUTPUT CODE, VALUE EXPECTED IN C
; A,FLAGS,C MODIFIED
;
3C32      CO:
3C32 3A0300  LDA     IOBYT        ; CONSOLE OUTPUT
3C35 E603      ANI     NOT CMSG   ; GET STATUS BYTE
3C37 C2463C  JNZ     COO        ; GET CONSOLE BITS
3C3A      TTYOUT:
3C3A DB01      IN      TTS        ; TEST FOR CRT
3C3C E604      ANI     TTYBE      ; CONSOLE = TTY
3C3E C23A3C  JNZ     TTYOUT      ; LOOP UNTIL READY
3C41 79      MOV    A,C
3C42 2F      CMA
3C43 D300      OUT     TIO        ; OUTPUT CHARACTER
3C45 C9      RET                ; RETURN
3C46      COO:
3C46 FE01      CPI     CCRT       ; CONSOLE = CRT?
3C48 C2573C  JNZ     CO3        ; TEST FOR BATCH
3C4B      CRTOUT:
3C4B DB05      IN      CRTS      ; CONSOLE = CRT

```



```

3C4D E604 ANI CRTBE
3C4F C24B3C JNZ CRTOUT ; LOOP UNTIL READY
3C52 79 MOV A,C
3C53 2F CMA
3C54 D304 OUT CRT0
3C56 C9 RET
3C57 CO3:
3C57 FE02 CPI BATCH
3C59 CACC3D JZ LO ; BATCH MODE, OUTPUT = LIST
3C5C C30337 JMP COLOC ; BRANCH TO USER CONSOLE OUTPUT

; READ TWO ASCII CHARACTERS, DECODE INTO 8 BITS BINARY
;
; BYTE:
3C5F CALL RIX ; READ CHAR FROM TAPE
3C62 CD003E CALL NIBBLE ; CONVERT ASCII TO HEX
3C65 07 RLC
3C66 07 RLC
3C67 07 RLC
3C68 07 RLC ; SHIFT FOUR PLACES
3C69 4F MOV C,A
3C6A CDEE3E CALL RIX
3C6D CD003E CALL NIBBLE ; GET LOWER NIBBLE
3C70 B1 ORA C
3C71 4F MOV C,A
3C72 82 ADD D ; UPDATE CHECKSUM
3C73 57 MOV D,A
3C74 79 MOV A,C
3C75 C9 RET ; RETURN

; EXTERNALLY REFERENCED ROUTINE
; CONSOLE INPUT CODE, VALUE RETURNED IN A
; A, FLAGS MODIFIED
;
; CI:
3C76 LDA IOBYT ; CONSOLE INPUT
3C76 3A0300 ; GET STATUS BYTE
3C79 E603 ANI NOT CMSK ; GET CONSOLE BITS
3C7B C2893C JNZ CI1 ; TEST FOR CRT
3C7E TTYIN:
3C7E DB01 IN TTS ; TTY STATUS PORT
3C80 E601 ANI TTYDA ; CHECK FOR DATA AVAILABLE
3C82 C27E3C JNZ TTYIN
3C85 DB00 IN TTI ; READ THE CHARACTER
3C87 CIO:
3C87 2F CMA
3C88 C9 RET ; RETURN
3C89 CII:
3C89 FE01 CPI CCRT ; CONSOLE = CRT?
3C8B C29A3C JNZ CI2 ; TEST FOR BATCH
3C8E CRTIN:
3C8E DB05 IN CRTS ; CRT STATUS PORT

```

```

3C90 E601      ANI      CRTDA      ; CHECK FOR DATA AVAILABLE
3C92 C28E3C    JNZ      CRTIN      ; NOT READY, CONTINUE LOOPING
3C95 DB04      IN       CRTI       ; READ THE CHARACTER
3C97 C3873C    JMP      CIO
3C9A          CI2:
3C9A FE02      CPI      BATCH
3C9C CA943E    JZ       RI          ; BATCH MODE, INPUT = READER
3C9F C30037    JMP      CILOC      ; CONSOLE = USER DEVICE

;
; CONVERT 4 BIT HEX VALUE TO ASCII CHARACTER
;
3CA2          CONV:
3CA2 FE0A      CPI      10
3CA4 FAA93C    JM      CNO        ; LESS THAN 10, (0-9)
3CA7 C607      ADI      'A'-'0'-10 ; ADJUST OF (A-F)
3CA9          CNO:
3CA9 C630      ADI      '0'        ; ADD BIAS FOR ASCII
3CAB 4F        MOV      C,A
3CAC C9        RET              ; RETURN

;
; TYPE CARRIAGE RETURN AND LINE FEED ON CONSOLE
;
3CAD          CRLF:
3CAD 0E0D      MVI      C,CR      ; <CR>
3CAF CD323C    CALL     CD
3CB2          LFX:
3CB2 0E0A      MVI      C,LF      ; <LF>
3CB4 C3323C    JMP      CU

;
; EXTERNALLY REFERENCED ROUTINE
; CONSOLE INPUT STATUS CODE
; A, FLAGS MODIFIED
;
3CB7          CSTS:
3CB7 3A0300    LDA      IOBYT      ; CONSOLE INPUT STATUS
3CBA E603      ANI      NOT CMsk    ; GET STATUS BYTE
3CBC C2C43C    JNZ      CS0        ; CONSOLE = TTY?
3CBF DB01      IN       TTS        ; CONSOLE = CRT
3CC1 C3CB3C    JMP      CS1        ; GET TTY STATUS
3CC4          CS0:
3CC4 FE01      CPI      CCRT
3CC6 C2D23C    JNZ      CS3
3CC9 DB05      IN       CRTS      ; GET CRT STATUS
3CCB          CS1:
3CCB E601      ANI      TTYDA
3CCD 3E00      MVI      A,FALSE    ; RETURN FALSE IF NO DATA AVAILABLE
3CCF          CS2:
3CCF C0        RNZ
3CD0 2F        CMA
3CD1 C9        RET              ; RETURN
3CD2          CS3:

```



```

3CD2 FE02 CPI BATCH
3CD4 3EFF MVI A,TRUE
3CD6 CACF3C JZ CS2
3CD9 C31837 JMP CSLOC

; READ BNPF TAPE RECORD, BUILD BYTE, STORE IN MEMORY
; IF ERROR, ABORT COMMAND
;
3CDC
3CDC CDEE3E CALL RIX ; READ TAPE
3CDF FE42 CPI 'B' ; SCAN FOR 'B'
3CE1 C2DC3C JNZ DECODE
3CE4 3601 MVI M,1 ; INITIALIZE MEMORY
3CE6
3CE6 CDEE3E DC0: CALL RIX ; GET DATA
3CE9 FE4E CPI 'N' ; CHECK FOR 'N'
3CEB C2FD3C JNZ DC2 ; NO, CHECK FOR 'P'
; CARRY = 0

3CEE
3CEE 7E DC1: MOV A,M ; SHIFT IN DATA BIT
3CEF 17 RAL
3CF0 77 MOV M,A
3CF1 D2E63C JNC DC0 ; IF CARRY IS SET, 8 BITS READ
3CF4 CDEE3E CALL RIX ; TEST FOR REQ'D 'F'
3CF7 FE46 CPI 'F'
3CF9 C2203C JNZ LER
3CFC C9 RET ; RETURN
3CFD
3CFD C6B0 DC2: ADI -'P'
3CFF C2203C JNZ LER ; ERROR
3D02 C3EE3C JMP DC1 ; CARRY IS SET

; 2.0 MS DELAY
;
3D05
3D05 F5 DELAY: PUSH PSW
3D06 C5 PUSH B
3D07 3E14 NVI A,DLY
3D09 060C MVI B,12 ; 100 MICROSECOND INNER LOOP
3D0B
3D0B 48 DL0: MOV C,B
3D0C
3D0C 0D DL1: DCR C
3D0D C20C3D JNZ DL1
3D10 3D DCR A
3D11 C20B3D JNZ DLO
3D14 C1 POP B
3D15 F1 POP PSW
3D16 C9 RET ; RETURN

; CONVERT BINARY NUMBER TO A STRING OF ASCII DIGITS

```

```

; HL = BINARY NUMBER
; DE = DIVISOR (DESCENDING POWERS OF 10)
; B = LEADING ZERO SUPPRESSION CHARACTER
; A,C = TEMPORARIES
;
DIGIT:
3D17      OE30      MVI      C,'0'      ; INITIALIZE CHARACTER
3D19      7D        DG0:    MOV      A,L      ; SUB DENOM (DE) FROM NUMERATOR (HL)
3D1A      93        SUB      E
3D1B      6F        MOV      L,A
3D1C      7C        MOV      A,H
3D1D      9A        SBB      D
3D1E      67        MOV      H,A
3D1F      DA263D    JC      DG1      ; NEGATIVE RESULT, ALL DONE
3D22      0C        INR      C      ; COUNT NUMBER OF SUBTRACTS
3D23      C3193D    JMP      DG0
3D26      19        DG1:    DAD      D      ; ADJUST HL
3D27      79        MOV      A,C
3D28      FE30      CPI      '0'      ; CHECK FOR LEADING ZERO SUPPRESSION
3D2A      C2313D    JNZ      DG3
3D2D      48        MOV      C,B
3D2E      DG2:      JMP      PO      ; PUNCH CHARACTER
3D2E      C36C3E    DG3:      MVI      B,'0'
3D31      0630      JMP      DG2
3D33      C32E3D
;
; ENCODE A BPNF WORD AND PUNCH IT
;
ENCODE:
3D36      0E42      MVI      C,'B'      ; PUNCH A 'B'
3D38      CD6C3E    CALL      PO
3D3B      0608      MVI      B,8      ; 8 BIT COUNT
3D3D      7E        MOV      A,M      ; GET DATA
3D3E      EN0:
3D3E      07        RLC      ; ROTATE TO SET CARRY
3D3F      F5        PUSH     PSW      ; SAVE INTERMEDIATE RESULT
3D40      3E00      MVI      A,0      ; COMPUTE EITHER 'P' OR 'N'
3D42      17        RAL      ; BASED ON FOLLOWING ALGORITHM:
3D43      17        RAL      ; CHAR = 'N' + 2*CARRY
3D44      C64E      ADI      'N'      ; CHAR = 'N' IF CARRY = 0
3D46      4F        MOV      C,A      ; CHAR = 'P' IF CARRY = 1
3D47      CD6C3E    CALL      PO
3D4A      F1        POP      PSW
3D4B      05        DCR      B
3D4C      C23E3D    JNZ      EN0
3D4F      0E46      MVI      C,'F'
3D51      CD6C3E    CALL      PO
3D54      0E20      MVI      C,' '

```



```

3D56 C36C3E      JMP      PO
;
; EVALUATE EXPRESSION: <EXPR>,<EXPR>,<EXPR>
;
3D59            EXPR:
3D59 210000      LXI      H,0          ; INITIAL VALUE OF PARAMETER
3D5C            EX0:
3D5C CD6D3F      CALL     TI          ; GET A CHARACTER
3D5F            EX1:
3D5F 47          MOV      B,A          ; SAVE DELIMITER CHARACTER
3D60 CD003E      CALL     NIBBLE       ; CONVERT TO HEX
3D63 DA6F3D      JC       EX2         ; NOT LEGAL CHAR, TREAT AS DELIMITER
3D66 29          DAD      H           ; *2
3D67 29          DAD      H           ; *4
3D68 29          DAD      H           ; *8
3D69 29          DAD      H           ; *16
3D6A B5          ORA      L
3D6B 6F          MOV      L,A
3D6C C35C3D      JMP      EXO         ; GET ANOTHER CHARACTER
3D6F            EX2:
3D6F E3          XTHL
; GET RETURN ADDRESS OFF STACK
; PUT HL ON
; REPLACE RETURN ADDRESS
3D70 E5          PUSH     H
3D71 78          MOV      A,B
3D72 CD493E      CALL     P2C         ; TEST DELIMITER CHARACTER
3D75 D27D3D      JNC      EX3
3D78 0D          DCR      C           ; CR ENTERED
3D79 C2203C      JNZ      LER         ; TOO FEW PARAMS
3D7C C9          RET
3D7D            EX3:
3D7D C2203C      JNZ      LER         ; ILLEGAL DELIMITER
3D80 0D          DCR      C
3D81 C2593D      JNZ      EXPR
3D84 C9          RET
3D85            EXF:
3D85 0E01        MVI      C,1         ; ENTRY POINT FOR CONDITIONAL PARAMETERS
3D87 210000      LXI      H,0
3D8A C35F3D      JMP      EX1
;
; COMPARE HL WITH DE:
; IF HL < DE THEN CARRY = 0;
; IF HL = DE THEN CARRY = 0;
; IF HL > DE THEN CARRY = 1;
;
3D8D            HILO:
3D8D 23          INX      H           ; BUMP HL
3D8E 7C          MOV      A,H         ; TEST FOR HL = 0
3D8F B5          ORA      L
3D90 37          STC
3D91 C8          RZ
3D92 78          MOV      A,E         ; DE = HL, SET/RESET CARRY

```

when called c reg has no. of param. saved

```

3D93 95          SUB      L
3D94 7A          MOV      A,D
3D95 9C          SBB      H
3D96 C9          RET              ; RETURN
;
; CONVERT NIBBLE IN A-REGISTER TO ASCII IN A-REGISTER
; AND PRINT ON TELEPRINTER
;
3D97          HXD;
3D97 CDA23C      CALL      CONV
3D9A C3323C      JMP      CO
;
; EXTERNALLY REFERENCED ROUTINE
; I/O SYSTEM STATUS CODE
; STATUS BYTE RETURNED IN A
;
3D9D          IOCHK:
3D9D 3A0300      LDA      IOBYT      ; GET STATUS BYTE
3DA0 C9          RET              ; RETURN
;
; EXTERNALLY REFERENCED ROUTINE
; SET I/O CONFIGURATION
; VALUE EXPECTED IN C
;
3DA1          IOSET:
3DA1 E5          PUSH     H          ; SAVE HL
3DA2 210300      LXI      H,IOBYT    ; POINT HL AT IOBYT
3DA5 71          MOV      M,C
3DA6 E1          POP      H          ; RESTORE HL
3DA7 C9          RET              ; RETURN
;
; PRINT CONTENTS OF HL IN HEX ON CONSOLE DEVICE
;
3DA8          LADR:
3DA8 7C          MOV      A,H          ; PRINT MSB
3DA9 CDB03D      CALL     LBYTE
3DAC 7D          MOV      A,L          ; PRINT LSB
3DAD C3B03D      JMP      LBYTE
;
; LIST A BYTE AS 2 ASCII CHARACTERS
;
3DB0          LBYTE:
3DB0 F5          PUSH     PSW          ; SAVE A COPY OF A
3DB1 0F          RRC
3DB2 0F          RRC
3DB3 0F          RRC
3DB4 0F          RRC
3DB5 E60F      ANI      0FH          ; UPPER 4 BITS
3DB7 CD973D      CALL     HXD
3DBA F1          POP      PSW          ; RETRIEVE ORIGINAL VALUE
3DBB E60F      ANI      0FH          ; LOWER 4 BITS

```



```

3DBD C3973D JMP HXD
;
; PUNCH 6 INCHES OF LEADER
;
3DC0 LEAD:
3DC0 063C MVI B,60 ; SET TO PUNCH 6 INCHES OF NULLS
3DC2 LEO:
3DC2 0E00 MVI C,0
3DC4 CD6C3E CALL PO
3DC7 05 DCR B
3DC8 C2C23D JNZ LEO
3DCB C9 RET ; RETURN
;
; EXTERNALLY REFERENCED ROUTINE
; LIST OUTPUT CODE
; VALUE EXPECTED IN C
;
3DCC LO: ; LIST OUTPUT
3DCC 3A0300 LDA IOBYT ; GET STATUS BYTE
3DCF E6C0 ANI NOT LMSK ; GET LIST BITS
3DD1 CA3A3C JZ TTYOUT ; LIST = TTY
3DD4 FE40 CPI LCRT
3DD6 CA4B3C JZ CRTOUT ; LIST = CRT
3DD9 FE80 CPI LUSE1 ; TEST FOR USER DEFINED LIST DEVICES
3DDB CA1237 JZ L1LOC ; BRANCH TO USER DEVICES
3DDE C31537 JMP L2LOC
;
; EXTERNALLY REFERENCED ROUTINE
; RETURN ADDRESS OF END OF MEMORY TO USER
; VALUE RETURNED IN (B,A)
;
3DE1 MEMCK:
3DE1 E5 PUSH H
3DE2 CDEA3D CALL MEMSIZ
3DE5 44 MOV B,H
3DE6 3EC0 MVI A,0C0H
3DE8 E1 POP H
3DE9 C9 RET
;
; FIND END OF MEMORY, SET STACK
;
3DEA MEMSIZ:
3DEA C5 PUSH B ; SAVE BC
3DEB 210000 LXI H,0 ; FIND END OF MEMORY
3DEE MEMO:
3DEE 46 MOV B,M ; FETCH CONTENTS OF MEMORY
3DEF 36AA MVI M,0AAH ; ATTEMPT TO WRITE INTO MEMORY
3DF1 7E MOV A,M ; NOW READ IT
3DF2 70 MOV M,B ; REPLACE ORIGINAL VALUE
3DF3 24 INR H
3DF4 FEAA CPI 0AAH ; IS LOCATION READ/WRITE?

```

```

3DF6 CAEE3D      JZ      MEMO          ; YES, CONTINUE
3DF9 25          DCR      H            ; POINT TO FIRST NON-RAM LOCATION
3DFA 01EEFF      LXI     B,EXIT-ENDX   ; COMPUTE TOP OF NEW STACK
3DFD 09          DAD      B
3DFE C1          POP      B            ; RESTORE BC
3DFF C9          RET                  ; RETURN

```

```

;
; DECODE ASCII CHAR IN A-REGISTER INTO HEX DIGIT IN A-REGISTER
;

```

```

3E00          NIBBLE:
3E00 D630      SUI      '0'
3E02 D8        RC
3E03 C6E9      ADI      '0'-'G'
3E05 D8        RC
3E06 C606      ADI      6
3E08 F20E3E    JP      N10
3E0B C607      ADI      7
3E0D D8        RC
3E0E          N10:
3E0E C60A      ADI      10
3E10 B7        ORA      A
3E11 C9          RET                  ; RETURN

```

```

;
; DISREGARD NOISE CHARACTERS
;

```

```

3E12          NOISE:
3E12 CD6D3F    CALL     TI
3E15 FE3D      CPI      'E'
3E17 C2123E    JNZ      NOISE
3E1A          NOO:
3E1A CD6D3F    CALL     TI
3E1D FE20      CPI      ' '
3E1F CA1A3E    JZ       NOO
3E22 C9          RET                  ; RETURN

```

```

;
; PUNCH CONTENTS OF HL IN HEX ON PUNCH DEVICE
;

```

```

3E23          PADR:
3E23 7C        MOV      A,H
3E24 CD2B3E    CALL     PBYTE
3E27 7D        MOV      A,L
3E28 C32B3E    JMP      PBYTE

```

```

;
; PUNCH A BYTE AS 2 ASCII CHARACTERS
;

```

```

3E2B          PBYTE:
3E2B F5        PUSH     PSW
3E2C 0F        RRC
3E2D 0F        RRC
3E2E 0F        RRC
3E2F 0F        RRC

```



```

3E30 E60F      ANI      OFH
3E32 CDA23C    CALL     CONV
3E35 CD6C3E    CALL     PO
3E38 F1        POP      PSW
3E39 F5        PUSH     PSW
3E3A E60F      ANI      OFH
3E3C CDA23C    CALL     CONV
3E3F CD6C3E    CALL     PO
3E42 F1        POP      PSW
3E43 82        ADD      D
3E44 57        MOV      D,A
3E45 C9        RET              ; RETURN

;
; TEST FOR NULL INPUT PARAMETER
;
3E46          PCHK:
3E46 CD6D3F    CALL     TI              ; GET A CHARACTER
3E49          P2C:
3E49 FE20      CPI      ' '
3E4B C8        RZ
3E4C FE2C      CPI      ','
3E4E C8        RZ
3E4F FE0D      CPI      CR
3E51 37        STC
3E52 3F        CMC
3E53 C0        RNZ
3E54 37        STC
3E55 C9        RET

;
; 520 MS DELAY FOR 1702A PROGRAMMING
;
3E56          PDLY:
3E56 C5        PUSH     B
3E57 06FF      MVI      B,LDLY
3E59          PDO:
3E59 CD053D    CALL     DELAY
3E5C 05        DCR      B
3E5D C2593E    JNZ      PDO
3E60 C1        POP      B
3E61 C9        RET              ; RETURN

;
; PUNCH CR,LF
;
3E62          PEO:
3E62 0E0D      MVI      C,CR
3E64 CD6C3E    CALL     PO
3E67 0E0A      MVI      C,LF
3E69 C36C3E    JMP      PO

;
; EXTERNALLY REFERENCED ROUTINE
; PUNCH OUTPUT CODE, VALUE EXPECTED IN C

```

```

; A, FLAGS, AND C MODIFIED
;
3E6C          POI:          LDA      IOBYT      ; PUNCH OUTPUT
3E6C          3A0300      ANI      NOT PMSK    ; GET STATUS BYTE
3E6F          E630        JZ       TTYOUT     ; GET PUNCH BITS
3E71          CA3A3C      JZ       TTYOUT     ; NO, PUNCH = TTY
3E74          FE10        CPI      PPTP      ; TEST FOR PTP
3E76          C28C3E      JNZ      POI        ; TEST FOR USER DEVICE(S)
3E79          POO:          IN       PTPS      ; PUNCH = PTP
3E79          DB01        ANI      PRDY      ; GET STATUS
3E7B          E640        JZ       POO        ; CHECK STATUS
3E7D          CA793E      MOV      A,C        ; LOOP UNTIL READY
3E80          79          OUT      PTPO
3E81          D303        MVI      A,PTPGO    ; START PUNCH
3E83          3E0A        OUT      PTPC
3E85          D301        MVI      A,PTPNO    ; STOP PUNCH
3E87          3E08        OUT      PTPC
3E89          D301        RET
3E8B          C9
3E8C          POI:          CPI      PUSE1
3E8C          FE20        JZ       P1LOC
3E8E          CA0C37      JMP      P2LOC
3E91          C30F37

; EXTERNALLY REFERENCED ROUTINE
; READER INPUT CODE
; VALUE RETURNED IN A, FLAGS MODIFIED
;
3E94          RI:          PUSH     H          ; READER INPUT
3E94          E5          LXI      H,IOBYT    ; SAVE HL
3E95          210300      MOV      A,M        ; POINT HL AT IOBYT
3E98          7E          ANI      NOT RMSK    ; READER = PTR?
3E99          E60C        JNZ      RI3        ; BRANCH TO PTR ROUTINE
3E9B          C2C03E      MVI      A,TTYGO    ; READER = TTY
3E9E          3E09        OUT      TTC
3EA0          D301        MVI      A,TTYNO
3EA2          3E08        OUT      TTC
3EA4          D301        MVI      H,60       ; SET TIMER
3EA6          263C
3EA8          RIO:          IN       TTS
3EA8          DB01        ANI      TTYDA
3EAA          E601        JZ       RI2        ; DATA IS READY
3EAC          CABA3E      CALL     DELAY      ; DELAY 2.0 MS
3EAF          CD053D      DCR      H
3EB2          25          JNZ      RIO
3EB3          C2A83E      XRA      A
3EB6          PI:          STC
3EB6          AF          POP      H          ; SET CARRY INDICATING EOF
3EB7          37          RET
3EB8          E1
3EB9          C9          ; RETURN

```



```

3E8A      DB00      RI2:      IN          TTI
3E8B      2F                CMA
3E8C      B7                ORA          A          ; CLEAR CARRY
3E8D      E1                POP          H
3E8E      C9                RET              ; RETURN
3E8F                        RI3:      ; PTR ROUTINE
3EC0      FE04      CPI          RPTR
3EC2      C2E53E     JNZ          R16
3EC5      3E0C      MVI          A,PTRGO      ; START PTR
3EC7      D301      OUT          PTRC
3EC9      3E08      MVI          A,PTRNO      ; STOP PTR
3ECB      D301      OUT          PTRC
3ECD      267F      MVI          H,7FH        ; SET TIMER TO MAX 250 MS.
3ECF                        RI4:
3ECF      DB01      IN          PTRS
3ED1      E620      ANI          PTRDA
3ED3      C2E03E     JNZ          R15
3ED6      CD053D     CALL        DELAY
3ED9      25                DCR          H
3EDA      C2CF3E     JNZ          R14
3EDD      C3B63E     JMP          R11
3EE0                        RI5:
3EE0      DB03      IN          PTRI          ; GET THE DATA
3EE2      B7                ORA          A
3EE3      E1                POP          H
3EE4      C9                RET              ; RETURN
3EE5                        RI6:
3EE5      E1                POP          H
3EE6      FE08      CPI          RUSE1
3EE8      CA0637     JZ          R1LOC
3EEB      C30937     JMP          R2LOC

; GET CHARACTER FROM READER, MASK OFF PARITY BIT
;
3EEE      CD943E     CALL        RI
3EEF      DA203C     JC          LER
3EF4      E67F      ANI          7FH
3EF6      C9                RET              ; RETURN

; RESTART 1 CODE
; (PROGRAMMED BREAKPOINT)
;
3EF7      RESTART:
3EF7      E5                PUSH        H          ; SAVE MACHINE STATE
3EF8      D5                PUSH        D
3EF9      C5                PUSH        B
3EFA      F5                PUSH        PSW
3EFB      CDEA3D     CALL        MEMSIZ      ; HL = NEW STACK POINTER
3EFE      EB                XCHG

```

```

      1      +      FETCH      10      ; COMPUTE ORIGINAL STACK POINTER
3EFF 1 210A00 +      LXI      H,0000AH      ; IN THE STACK
3F02 1 39      +      DAD      SP
3F03      0604      MVI      B,4      ; COUNT FOR TRANSFER OF MACHINE STATE
                                   ; TO STORAGE (MOVE THE STACK)

3F05      EB      XCHG
3F06      RST0:    DCX      H
3F06      2B      MOV      M,D
3F07      72      DCX      H
3F08      2B      MOV      M,E
3F09      73      POP      D
3F0A      D1      DCR      B
3F0B      05      JNZ      RST0
3F0C      C2063F  POP      B
3F0F      C1      DCX      B
3F10      0B      SPHL
3F11      F9      ; GET OLD PC = B,C; OLD HL = D,E
                                   ; DECREMENT TO POINT AT TRAPPED CODE
                                   ; NEW STACK VALUE

      1      +      FETCH      TLOC
3F12 1 211400 +      LXI      H,TLOC      ; IN THE STACK
3F15 1 39      +      DAD      SP
3F16      7E      MOV      A,M
3F17      91      ; TEST IF THIS IS A PROGRAMMED RESTART
3F18      23      SUB      C      ; OR A CONSOLE RESTART
3F19      C2213F  INX      H
3F1C      7E      JNZ      RST1
3F1D      90      MOV      A,M
3F1E      CA2F3F  SUB      B
3F21      RST1:   JZ       RST3
3F21      23      INX      H
3F22      23      INX      H
3F23      7E      MOV      A,M
3F24      91      SUB      C
3F25      C22E3F  JNZ      RST2
3F28      23      INX      H
3F29      7E      MOV      A,M
3F2A      90      SUB      B
3F2B      CA2F3F  JZ       RST3
3F2E      RST2:   INX      B
3F2E      03      RST3:
3F2F      1      +      FETCH      LLOC
3F2F 1 210F00 +      LXI      H,LLOC      ; IN THE STACK
3F32 1 39      +      DAD      SP
3F33      73      MOV      M,E
3F34      23      INX      H
3F35      72      MOV      M,D      ; SAVE OLD HL
3F36      23      INX      H
3F37      23      INX      H
3F38      71      MOV      M,C      ; SAVE OLD PC
3F39      23      INX      H
3F3A      70      MOV      M,B

```

```

3F3B C5          PUSH B
3F3C 0E2A        MVI C,'*'
3F3E CD323C      CALL CO
3F41 E1          POP H          ; RETRIEVE OLD PC FOR DISPLAY
3F42 CDA83D      CALL LADR      ; DISPLAY PC
1          +     FETCH TLOC     ; CLEAR TRAPS
3F45 1 211400 +   LXI H,TLOC    ; IN THE STACK
3F48 1 39        +   DAD SP
3F49 1602        MVI D,2        ; SET COUNT FOR TWO TRAPS
3F4B          RST4:
3F4B 4E          MOV C,M        ; GET LSB OF ADDRESS
3F4C 3600        MVI M,0        ; CLEAR MEMORY
3F4E 23          INX H
3F4F 46          MOV B,M        ; GET MSB OF ADDRESS
3F50 3600        MVI M,0
3F52 23          INX H
3F53 79          MOV A,C
3F54 B0          ORA B          ; TEST FOR VALID TRAP
3F55 CA5A3F      JZ RST5        ; ADDRESS = 0, NO TRAP TO RESTORE
3F58 7E          MOV A,M        ; GET OPCODE BYTE
3F59 02          STAX B         ; REPLACE IT
3F5A          RST5:
3F5A 23          INX H          ; POINT TO NEXT TRAP ADDRESS
3F5B 15          DCR D
3F5C C24B3F      JNZ RST4       ; REPEAT FOR TRAP 2
3F5F C36A38      JMP START

;
; SCAN TO END OF LINE
;
3F62          SCANOUT:
3F62 CD6D3F      CALL TI
3F65 FE0D        CPI CR
3F67 C2623F      JNZ SCANOUT
3F6A C3B23C      JMP LFX

;
; INPUT FROM CONSOLE, ECHOED AND RETURNED IN A
;
3F6D          TI:
3F6D CD763C      CALL C1
3F70 E67F        ANI 7FH
3F72 C5          PUSH B
3F73 4F          MOV C,A
3F74 CD323C      CALL CO
3F77 79          MOV A,C
3F78 C1          POP B
3F79 C9          RET          ; RETURN

;
; I/O SYSTEM PHYSICAL DEVICE TABLES
; 2 BYTES/ENTRY
; BYTE 0 = IDENTIFYING CHARACTER
; BYTE 1 = DEVICE SELECT BIT PATTERN

```



```

3F7A      ,
3F7A      ICT:      DB      'T',CTTY      ; CONSOLE = TTY
3F7C      5400      DB      'C',CCRT      ; CONSOLE = CRT
3F7E      4301      DB      'B',BATCH     ; BATCH MODE CONSOLE = READ,LIST
3F80      4202      DB      '1',CUSE      ; USER DEFINED CONSOLE DEVICE
3F82      3103      IRT:      DB      'T',RTTY      ; READER = TTY
3F82      5400      DB      'P',RPTR      ; READER = PTR
3F84      5004      DB      '1',RUSE1     ; USER DEFINED READER DEVICE 1
3F86      3108      DB      '2',RUSE2     ; USER DEFINED READER DEVICE 2
3F88      320C      OPT:      DB      'T',PTTY      ; PUNCH = TTY
3F8A      5400      DB      'P',PPTP      ; PUNCH = PTP
3F8C      5010      DB      '1',PUSE1     ; USER DEFINED PUNCH DEVICE 1
3F8E      3120      DB      '2',PUSE2     ; USER DEFINED PUNCH DEVICE 2
3F90      3230      OLT:      DB      'T',LTTY      ; LIST = TTY
3F92      5400      DB      'C',LCRT      ; LIST = CRT
3F94      4340      DB      '1',LUSE1     ; USER DEFINED LIST DEVICE 1
3F96      3180      DB      '2',LUSE2     ; USER DEFINED LIST DEVICE 2
3F98      32C0

```

```

;
; EXIT CODE TEMPLATE, TO BE EXECUTED IN RAM
;

```

```

;      DB      E      POP D      ; MONITOR WORK STACK ORIGIN
;      DB      D
;      DB      C      POP B
;      DB      B
;      DB      FLAGS POP PSW
;      DB      A
;      DB      SPL      POP H
;      DB      SPU      SPHL
;

```

```

3F9A      EXIT:      ; MONITOR STACK ORIGIN
3F9A      D1      POP      D      ; RESTORE D,E
3F9B      C1      POP      B      ; RESTORE B,C
3F9C      F1      POP      PSW    ; RESTORE A AND FLAGS
3F9D      E1      POP      H      ; RESTORE ORIGINAL STACK VALUE
3F9E      F9      SPHL
3F9F      FB      EI      ; ENABLE INTERRUPTS
3FA0      210000   LXI      H,$-S  ; RESTORE H,L
3FA1      HLX     EQU     $-2
3FA3      C30000   JMP      $-8    ; RETURN TO INTERRUPTED CODE
3FA4      PCX     EQU     $-2
3FA6      0000     T1A:    DW      0      ; TRAP 1 ADDRESS
3FA8      00      DW      0      ; TRAP 1 VALUE
3FA9      0000     DW      0      ; TRAP 2 ADDRESS
3FAB      00      DW      0      ; TRAP 2 VALUE
3FAC

```

```

ENDX:

```

```

; DISPLACEMENT OF REGISTER LOCATION FROM SP (LEVEL 0)

```

```

0005      ALOC EQU 5
0003      BLOC EQU 3
0002      CLOC EQU 2
0001      DLOC EQU 1
0000      ELOC EQU 0
0004      FLOC EQU 4
0010      HLOC EQU HLX-EXIT+9
000F      LLOC EQU HLX-EXIT+8
0013      PLOC EQU PCX-EXIT+9
0007      SLOC EQU 7
0014      TLOC EQU T1A-EXIT+8
;
; TABLE FOR ACCESSING REGISTERS
; TABLE CONTAINS:
;   (1) REGISTER IDENTIFIER
;   (2) STACK POINTER DISPLACEMENT
;   (3) PRECISION
;
ACTBL:
3FAC      410501      DB      'A',      ALOC,      1
3FAF      420301      DB      'B',      BLOC,      1
3FB2      430201      DB      'C',      CLOC,      1
3FB5      440101      DB      'D',      DLOC,      1
3FB8      450001      DB      'E',      ELOC,      1
3FBB      460401      DB      'F',      FLOC,      1
3FBE      481001      DB      'H',      HLOC,      1
3FC1      4C0F01      DB      'L',      LLOC,      1
3FC4      4D1002      DB      'M',      HLOC,      2
3FC7      501302      DB      'P',      PLOC,      2
3FCA      530702      DB      'S',      SLOC,      2
3FCD      FF          DB      OFFH
; TABLE TERMINATOR
;
; END OF PROGRAM
;
      END

```

NO PROGRAM ERRORS

SYMBOL TABLE

* 01

A	0007	ACTBL	3FAC	ALOC	0005	ASO	38F8
AS1	3905	AS2	3912	ASSIG	38C3	B	0000
BATCH	0002	BEGIN	3827	BGO	382E	BG1	383F
BLK	3C30	BLOC	0003	BNO	392A	BN1	3956
BNPF	391F	BYTE	3C5F	C	0001	CCRT	0001
CI	3C76	CIO	3C87	CI1	3C89	CI2	3C9A
CILOC	3700	CLOC	0002	CM0	396F	CM1	3993
CMSK	00FC	CNO	3CA9	CO	3C32	CO0	3C46
CO3	3C57	COLUC	3703	COMP	3968	CONV	3CA2
CR	000D	CRLF	3CAD	CRTBL	0004	CRTDA	0001
CRTI	0004	CRTIN	3C8E	CRT0	0004	CRT0U	3C4B
CRTS	0005	CS0	3CC4	CS1	3CCB	CS2	3CCF
CS3	3CD2	CSLOC	3718	CST5	3CB7	CTTY	0000
CUSE	0003	D	0002	DC0	3CE6	DC1	3CEE
DC2	3CFD	DEBUG	0000	DECOD	3CDC	DELAY	3D05
DGO	3D19	DG1	3D26	DG2	3D2E	DG3	3D31
DIO	39A0	DI1	39A6	DIGIT	3D17	DISP	399B
DLO	3D0B	DL1	3D0C	DLOC	0001	DLY	0014
DSB	0008	E	0003	ELOC	0000	ENO	3D3E
ENB	0000	ENCOD	3D36	ENDX	3FAC	EOF	39BC
EX0	3D5C	EX1	3D5F	EX2	3D6F	EX3	3D7D
EXF	3D85	EXIT	3F9A	EXPR	3D59	FALSE	0000
FETCH	03E1	FIO	39E5	FILL	39DE	FIRST	0000
FLOC	0004	GO0	3A09	GO1	3A0F	GO2	3A27
GO3	3A31	GOTO	39EF	H	0004	HEXN	3A39
HILO	3D8D	HLOC	0010	HLX	3FA1	HXD	3D97
ICT	3F7A	INIT	0000	IOBYT	0003	IOCHK	3D9D
IOMOD	38D0	IOSET	3DA1	IRT	3F82	L	0005
L1LOC	3712	L2LOC	3715	LADR	3DA8	LBYTE	3DB0
LCRT	0040	LDLY	00FF	LE0	3DC2	LEAD	3DC0
LER	3C20	LF	000A	LFx	3CB2	LLOC	000F
LMSK	003F	LO	3DCC	LO0	3A5E	LOAD	3A56
LTTY	0000	LUSE1	0080	LUSE2	00C0	LVER	0009
M	0006	MEM0	3DEE	MEMCK	3DE1	MEMSI	3DEA
MODIO	032D	MOVE	3A6A	MVO	3A71	NIO	3E0E
NIBBL	3E00	NO0	3E1A	NOISE	3E12	NULL	3A7D
OLT	3F92	OPT	3F8A	P1LOC	370C	P2C	3E49
P2LOC	370F	PAD	0002	PADR	3E23	PBITA	0080
PBYTE	3E2B	PCHK	3E46	PCMD	0002	PCX	3FA4
PD0	3E59	PDI	0002	PDLY	3E56	PDO	0003
PEOL	3E62	PLOC	0013	PMSK	00CF	PO	3E6C
PO0	3E79	PU1	3E8C	PPTP	0010	PRO	3A8D
PR1	3A8F	PR2	3AD5	PRDY	0040	PROG	3A83
PROGO	0080	PROMC	0001	PRONO	0000	PSW	0006
PTPC	0001	PTPGO	000A	PTPNO	0008	PTPO	0003
PTPS	0001	PTRC	0001	PTRDA	0020	PTRGO	000C
PTRI	0003	PTRNO	0008	PTR6	0001	PTTY	0000
PUSE1	0020	PUSE2	0030	R1LOC	3706	R2LOC	3709
RBIT	0001	RCMD	0004	READ	3ADF	RED0	3AE3
RED1	3B03	RED2	3B15	RED3	3B22	RESTA	3EF7
RI	3E94	RI0	3EA8	RI1	3EB6	RI2	3EBA
RI3	3EC0	RI4	3ECF	RI5	3EE0	RI6	3EE5

RIX	3EEE	RMSK	00F3	RPTR	0004	RS1	0008
RST0	3F06	RST1	3F21	RST2	3F2E	RST3	3F2F
RST4	3F4B	RST5	3F5A	RTTY	0000	RUSE1	0008
RUSE2	000C	SCAND	3F62	SLOC	0007	SP	0006
START	386A	SU0	3B31	SU1	3B50	SUBS	3B26
T1A	3FA6	TBL	3893	TI	3F6D	TLOC	0014
TR0	3B5B	TRAN	3B54	TRUE	FFFF	TTC	0001
TTI	0000	TTO	0000	TTS	0001	TTYBE	0004
TTYDA	0001	TTYGO	0009	TTYIN	3C7E	TTYNO	0008
TTYOU	3C3A	VER	000A	VER0	3861	VERS	381E
WRIO	3B77	WRI1	3B8C	WRI2	3B8F	WRI3	3BAA
WRITE	3B6F	X	3B8C	X0	3BC2	X1	3BD3
X2	3BD6	X3	3BEE	X4	3C10	X5	3C12

* 02

TEST 38EB

* 03

* 04

* 05

* 06

* 07

* 08

* 09

* 10

* 11

* 12

